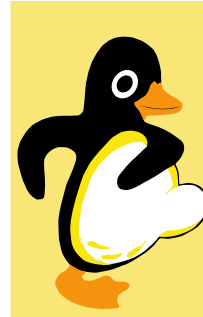


Genetische Algorithmen mit mehreren Zielfunktionen

Dr. Ralf Schlatterbeck
Open Source Consulting

Email: office@runtux.com
Web: http://www.runtux.com
Tel. +43/650/621 40 17



Genetische Algorithmen

- Suche einer optimalen Lösung für ein Problem
- Inspiriert von der Natur
- Gene sind Sequenzen von Zahlen oder Symbolen (binär, Integer, Floating-Point)
- Aus einem Gen wird eine potentielle Lösung für das Problem aufgebaut: Phänotyp
- Algorithmus arbeitet mit Population einer bestimmten Größe
- Fitness: Bewertungsfunktion wird optimiert
- Maximierung oder Minimierung möglich
- *Individuum*: Genotyp + Phänotyp



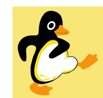
Skizze Algorithmus

1. Zufällige Initialisierung der Individuen
2. Bewertung der Individuen
3. Loop: Erzeugen neue Generation:
   (a) Selektion: Auswahl für nächsten Schritt
   (b) Anwendung genetische Operatoren auf selektierte Individuen
   (c) Bewertung der neuen Individuen
   (d) Erzeugen nächste Generation aus neuen und alten Individuen



Implementation in PGApack

- PGApack: Parallel Genetic Algorithm Package
- Open Source genetic algorithm implementation
- Originally from David Levine, Argonne National Laboratory
- Parallel execution via MPI (Message Passing Interface)
- Construction kit for your experiments
- I'm maintaining a fork since 2017
- Current version implements differential evolution and other variants
- PGApy is a python-wrapper (since 2005)



Variante Differential Evolution

Nur für Floating-Point Gene! [SP97, PSL05]

(a) Selektion: Nimm der Reihe nach jedes Individuum k

(b) Wähle zufällig drei Individuen, alle verschieden und verschieden von k

(c) Bilde Vektor-Differenz aus zwei dieser Individuen und addiere Differenz zum dritten

(d) Optional: Crossover des Resultats mit k

(e) nächste Generation: Das neue Individuum ersetzt das alte wenn es besser ist

Numerische Optimierung – auch für Antennen!

Randbedingungen

- Einfache „Box Constraints“
- Minimum und Maximum von jedem Parameter (=Gen)
- z.B. bei Längen: Sind nicht negativ
- . . . und meist ist Größenordnung bekannt (mm, m, oder km)
- Von praktisch allen genetischen Algorithmen unterstützt
- Komplizierte Bedingungen: mehrere Zielfunktionen
- Eval-Funktion hat mehrere Rückgabewerte

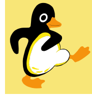


Mehrere Zielfunktionen

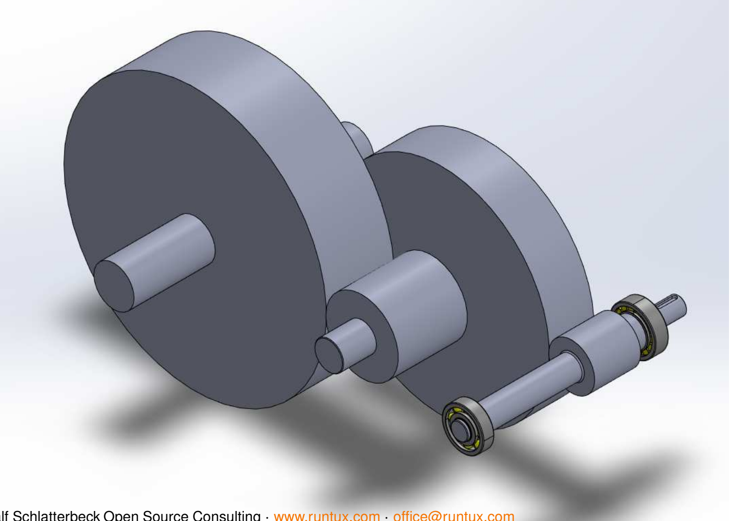
- Traditionell: Nur eine Zielfunktion (objective function)
- Wird maximiert oder minimiert
- Mehrere Zielfunktion:
  - für Randbedingungen (Constraints)
  - mehrere gleichzeitige Ziele

$\rightarrow$ Randbedingungen erschweren das Optimierungsproblem

$\rightarrow$ Ziele können sich widersprechen!



Beispiel: 2-Stufen Getriebe





Beispiel: 2-Stufen Getriebe



Quelle: Wikimedia Commons, CC-by-sa 4.0



Beispiel: Randbedingungen

Beispiel

- Vier Zahnräder
- Vorgegebenes Übersetzungsverhältnis
- Randbedingung: Zahnräder müssen sich gleichmäßig abnutzen

$\rightarrow$ Anzahl der Zähne von je einem kleinen und einem großen Rad ohne gemeinsamen Faktoren (GGT 1)

- $17 \leq$ Anzahl Zähne ≤ 90
- Verhältnis 950:150 = 19:3

Beispiel: Randbedingungen

- Lösung ohne Randbedingung: 17, 18, 38, 51
- Aber: 51 ist Vielfaches von 17
- und GGT (51, 18) = 3 und GGT (38, 18) = 2
- Lösung mit Randbedingung: 23, 49, 83, 86
- Abweichung vom vorgegebenen Verhältnis: 0.00467%
- gefunden mit PGApy
- Differential Evolution, gerundet auf ganze Zahlen

$\rightarrow$ `examples/gears.py`

- Neues Project *gearbox* mit mehr mechanischen Details

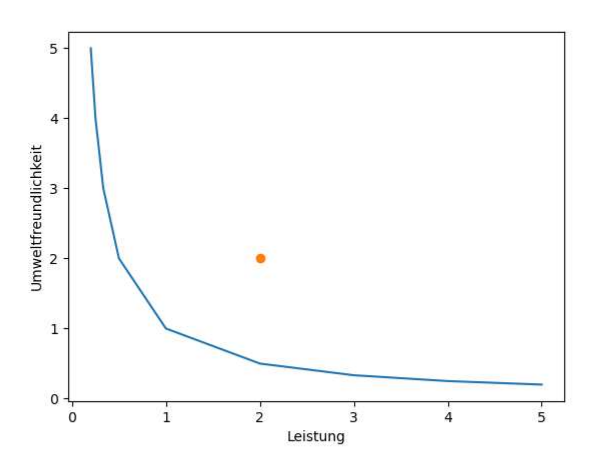


Mehrere Ziele

- Man kann alle Ziele in eine Funktion „vermantschgern“
- Das legt aber die Gewichtung von vorneherein fest
- Vergleiche Vortrag letztes Jahr
- Daher: Mehrere Ziele in *einem Optimierungslauf*
- Ziele werden sich typischerweise widersprechen
- Wir wollen also möglichst alle „guten“ Lösungen finden
- Und später entscheiden welche Kompromisse wir eingehen



Beispiel mehrere Ziele: Auto





Mehrere Ziele: Pareto Front

- Eine Lösung dominiert eine andere Lösung wenn
- ... sie in mindestens einem Kriterium besser ...
- ... und in keinem anderen Kriterium schlechter ist
- Alle nicht-dominierten Lösungen sind Pareto-Optimal
- ... und bilden eine Pareto-Front (graphisch)
- nach dem italienischen Wissenschaftler Vilfredo Pareto
- ... der das Konzept der Pareto-Optimalität definiert hat



NSGA-II

- Algorithmus von 2002 (!) Kalyanmoy Deb et. al [DPAM02]
- Non-dominated Sorting Genetic Algorithm
- Sortiert nach Dominanz-Beziehung
- Pareto-Front
- Alle nicht-dominierten Lösungen entfernen
- Zweite Front von nicht-dominierten Lösungen usw.
- Letzte Front wird nur teilweise übernommen
- Lösungen in Bereichen wo es wenige Lösungen gibt bevorzugt

$\rightarrow$ Crowding Metric

- Nach Konvergenz bleibt typisch eine Front



Beispiel: Antennen

- Simulation mit NEC (numerical electromagnetics code)
- Stammt aus den 80ern mit Lochkarten
- Rewrite in C/C++ „nec2c“
- nec2c: Übersetzt ein NEC file in Tabellen-Ausgabe
- xnec2: Nimmt ein NEC-File und zeigt Grafik
- xnecview nimmt nec2c Ausgabe und zeigt Grafik
- ... alle in Debian gepackaged

Antennen-Optimierung mit genetischen Algorithmen ist schon in den 90er Jahren publiziert [Lin97] aber vielleicht nicht genutzt weil bis 2015 patentiert? [AL95]



Antennen-Parameter: Stehwellenverhältnis

- Stehwellenverhältnis VSWR (Voltage Standing Wave Ratio)
- Wenn eine Antenne gut angepasst ist wird alles abgestrahlt
- Wenn nicht wird ein Teil reflektiert

→ Stehwelle

$$\text{SWR} = \frac{|U_V| + |U_R|}{|U_V| - |U_R|}$$

- Ist 1 wenn die Antenne gut angepasst ist
- Gute Werte < 2

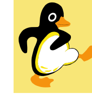


Antennen-Parameter: Gewinn

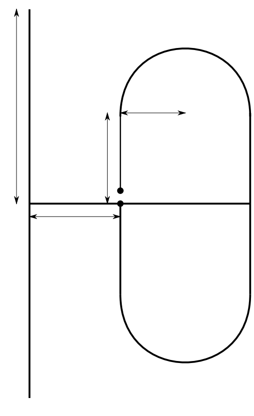
- Gewinn gegenüber einem idealen Kugelstrahler
- “Isotropic Radiator”
- Kann man nicht bauen aber ist gut als Referenz
- Einheit: dBi (Dezi-Bel über isotropic radiator)
- Eine Antenne mit Richtwirkung hat Gewinn > 0dB
- z.B. ein $\lambda/2$-Dipol hat 2.15 dBi Gewinn
- Gewinn gegenüber Dipol: dBd
- Umrechnung dBi $\rightarrow$ dBd: subtrahiere 2.15
- X dB entspricht einem Faktor $10^{X/10}$
- 3dB entspricht Faktor 2
- 10dB entspricht Faktor 10

Antennen-Parameter: Vorwärts Rückwärtsverhältnis

- Gewinn in Hauptrichtung
- . . . geteilt durch (max.) Gewinn in Rückwärtsrichtung
- In dB
- Intuitiv würde man annehmen das ist bei hohem Gewinn automatisch gegeben
- . . . aber Antennen mit hohem Gewinn haben oft ausgeprägte „Keulen“ in der Gegenrichtung

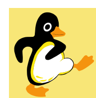


2-Elemente mit Faltdipol



- 4 Parameter
- Coding als Floating-Point

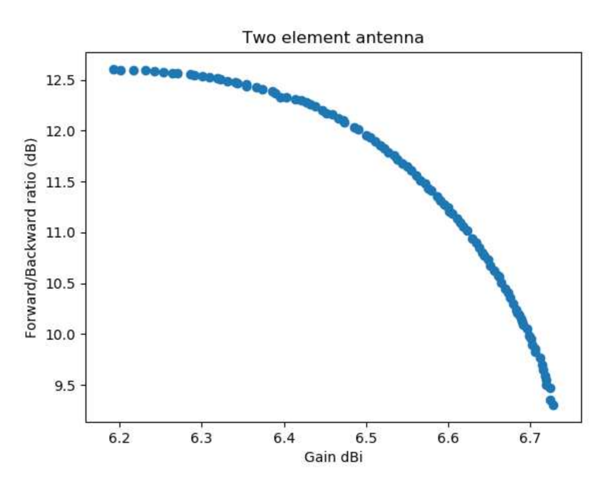
→ Differential Evolution



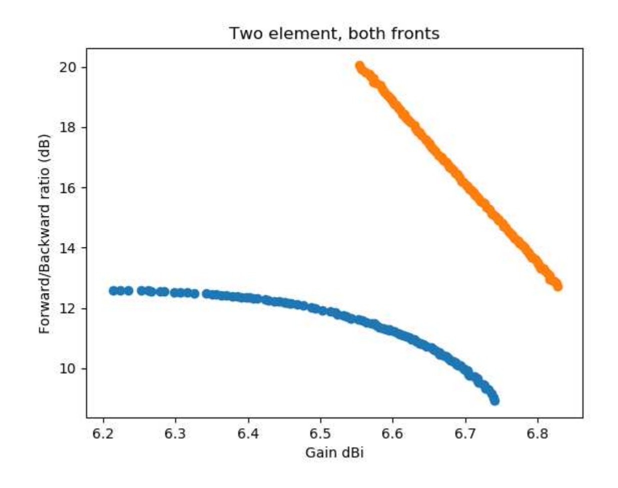
Antennen-Optimierung mit NSGA-II

- Eine Randbedingung: VSWR < 1.8
- Zwei Zielfunktionen: Gewinn und Vorwärts/Rückwärts
- Gen der Länge 4:
- Länge des Falt-Dipols
- Breite des Falt-Dipols
- Länge des zweiten Elements
- Abstand der Elemente
- Box-Constraints für alle Parameter

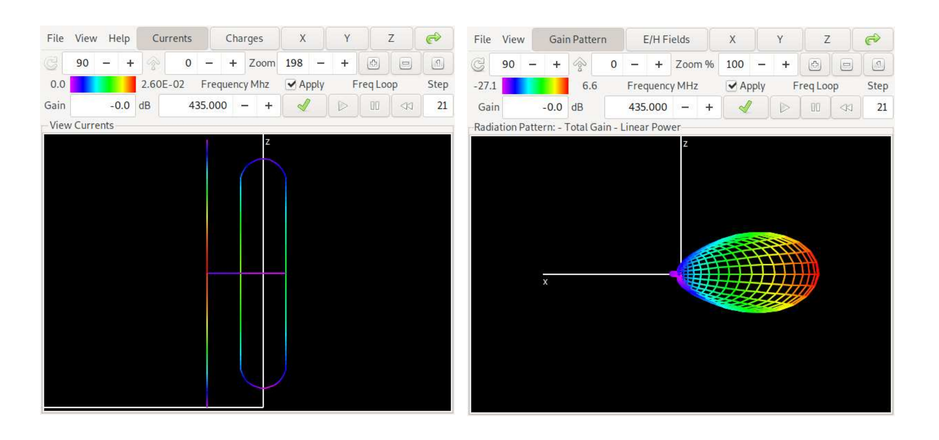
Pareto Front optimierte Antenne



Pareto Front anderer Optimierungslauf



Antenne Untere Front

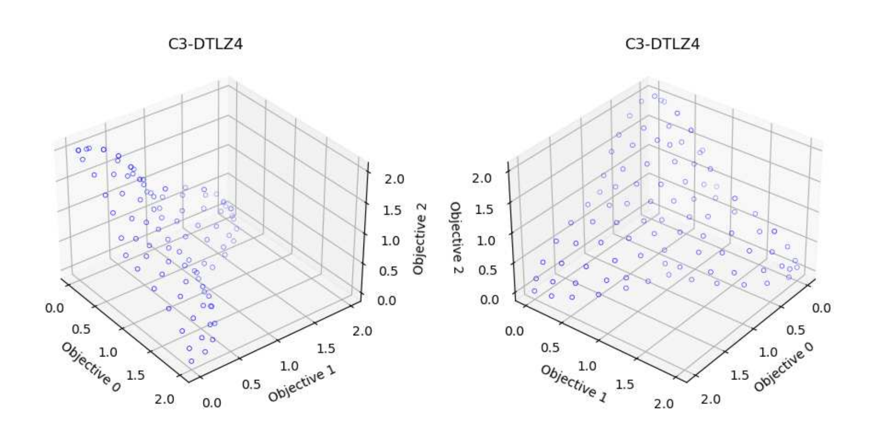


Antenne Obere Front

Ausblick: NSGA-III

- Publikation 2014 [DJ14] [JD14]
- Funktioniert mit mehr Zielfunktionen
- Größe der Pareto-Front steigt exponentiell mit Anzahl Zielfunktionen!
- . . . macht NSGA-II ineffizient
- NSGA-III: Beschränkung auf wenige Punkte
- Systematische Generierung von Ziel-Punkten
- Nur die Crowding-Metrik wurde ersetzt!
- Implementiert in PGAPack set 2021
- Bis jetzt nur synthetische (Test-)Probleme gelöst

NSGA-III Example





Software

- PGApack:
  github.com/schlatterbeck/pgapack
- PGApy:
  github.com/schlatterbeck/pgapy
- PyNEC:
  pypi.org/project/PyNEC/
- Antenna-Optimizer:
  github.com/schlatterbeck/antenna-optimizer
- Gearbox:
  github.com/schlatterbeck/gearbox



[AL95] Edward E. Altshuler and Derek S. Linden. Process for the design of antennas using genetic algorithms. US Patent US5719794A, July 1995.

[DJ14] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IE-*



EE Transactions on Evolutionary Computation, 18(4):577–601, August 2014.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[JD14] Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimizati-

on algorithm using reference-point-based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, August 2014.

[Lin97] Derek S. Linden. *Automated Design and Optimization of Wire Antennas Using Genetic Algorithms*. Dissertation, Massachusetts Institute of Technology, September 1997.

[PSL05] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, Heidelberg, 2005.

[SP97] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.